The pull method

The "pull method" can be used to verify if a fit of a data set was performed correctly and if data-associated uncertainties were biased.

If a random variable x is generated repeatedly with a Gaussian distribution of mean μ and width σ, then it is almost a tautology that the pull, defined as:

$g = (x - \mu) / \sigma$

will be distributed as a standard Gaussian with mean zero and unit width.

Thanks to the central limit theorem, this simple property can be applied in a wide range of situations from hypothesis testing to parameter estimation, where pulls provide evidence for various forms of bias and allow the verification of error coverage.

A pull distribution which mean is different from zero may indicate a systematic difference between the data and the theory (or fit).

If the sigma of the pull distribution is different from one then the uncertainties associated to the data could be under or over-estimated.

Goal of these exercises

The aim of these exercises is to check if a given function represents correctly a distribution of an observable determined using the detectors of the PAMELA apparatus.

Basic informations: you will work on a set of events coming out from a PAMELA experiment simulation. The given file

`/home/mocchiut/pamela/data/pamsimu2014.root`

contains the TTree pamcalotree, storing data with the PamCalo class, header file:

`/home/mocchiut/pamela/PamCalo/inc/PamCalo.h`

so library:

`/home/mocchiut/pamela/PamCalo/lib/Linux/libPamCalo.so`.

The ROOT file contains a mixture of different particles: protons, antiprotons, electrons and positrons in an energy range from about 1 to about 20 GeV.

In these exercises, we will select negatively charged particles from the sample (energy<0) and we will construct and save some variables on a different file. Then we will use a script to plot the observable distribution and we will calculate and draw the pull distribution in two different cases. Finally, we will fit the pull distributions with a Gaussian function.

Exercise 1

Write an an executable compiled program which reads the input file

`/home/mocchiut/pamela/data/pamsimu2014.root`

and gives as output a new ROOT file containing a TTree with two variables (a TBranch for each one):

- pID
- qtotene (≡qtot/energy)

Save into the new file the events for which the following condition is satisfied:

1. for each event “energy” is lower than zero.

Hints:

- to compile, remember to add also the compilation flags:

```
-I/home/mocchiut/pamela/PamCalo/inc
-L/home/mocchiut/pamela/PamCalo/lib/Linux/
-lPamCalo
```

- to run, remember to export LD_LIBRARY_PATH:

```
export LD_LIBRARY_PATH=/home/mocchiut/pamela/PamCalo/lib/Linux/:$LD_LIBRARY_PATH
```

Exercise 2

Write a ROOT-CINT script which reads the output file of exercise 2 (should be similar to this one: `/home/mocchiut/scripts/EM_output_230615.root` use this file if you are unable to complete or run exercise 1) and gives as output on the screen and on the disk (pdf format) a TCanvas, divided into two pads (one column, two rows – hint: `TCanvas::Divide`), which contains from top to bottom:

1. the distribution of qtotene for all events (`TH1D,X` range [-350.,-50.], number of bins: 300); overimpose a Gaussian function (called fgauss, use TF1) with constant value 5.38566e+01, mean -2.55415e+02 and sigma 3.88980e+01.

2. the distribution of qtotene for the events (another `TH1D,X` range [-350.,-50.], number of bins: 300) which pass the selection `pID==2` , overimpose the same Gaussian function of point 1.

Exercise 3

Update the script of exercise 2 in order to draw a new TCanvas divided into two pads (one columns, two rows – hint: `TCanvas::Divide`) which contain:

1. A TH1D (range [-5.,5.], 50 bins) with the distribution of the pulls for the histogram of exercise 1 point 1, where the pull are built bin per bin as:

   z = (bin_content – value_of_fgauss_at_bin_center) / (bin_content)$^{1/3}$.

   (hints: create a loop over the number of bins; remember to skip the overflow and underflow bins of TH1D; look at TH1D methods `GetBinContent`, `GetBinCenter`; look at `TF1::Eval` method).

   Fit the pull distribution with a new Gaussian function and print on the STDOUT the mean, the sigma and the chi2 of the fit.

2. A TH1D (range [-5.,5.], 50 bins) with the distribution of the pulls for the histogram of exercise 1 point 2 and the following definition of pull:

   z = (bin_content – value_of_fgauss_at_bin_center) / sqrt(bin_content)

   Fit the pull distribution with a new Gaussian function and print on the STDOUT the mean, the sigma and the chi2 of the fit.

Make the script checking which is the pull distribution with mean closer to zero and which is the pull distribution with sigma closer to one. Print the results on the STDOUT.

Save the canvas on the disk as pdf file.

Preparing the output

- create a directory and put inside this directory ALL the files you want me to correct and look at.
- create a README text file (named like EM_README.txt), inside the file write:
  - **your name and surname**
  - a list of the files you are submitting
  - **in details** how to compile and run the programs
  - any other comment and answer to question(s)

Timing and rules

- You have four hours time to do your work.
- You can search the web, look at manuals, look at any note you wrote during the course, etc.
- We will discuss what you have written at the oral examination on 2015/06/26 (25?), until that (if needed) you can change and improve your programs. In that case prepare an electronic version we can look at during the oral examination, we will compare it to the one handed in today and we will discuss any change and/or correction.